AD

AD-E403 300

Technical Report ARMET-TR-10010

PROJECTILE VELOCITY ESTIMATION USING AERODYNAMICS AND ACCELEROMETER MEASUREMENTS: A KALMAN FILTER APPROACH

Thomas Recchia

August 2010

U.S. ARMY ARMAMENT RESEARCH, DEVELOPMENT AND ENGINEERING CENTER

Munitions Engineering Technology Center

Picatinny Arsenal, New Jersey

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-01-0188*

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (*From - To*)
August 2010		November 2008

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
PROJECTILE VELOCITY ESTIMATION USING AERODYNAMICS AND ACCELEROMETER MEASUREMENTS: A KALMAN FILTER APPROACH	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHORS	5d. PROJECT NUMBER
Thomas Recchia	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army ARDEC, METC Munitions Systems & Technology Directorate (RDAR-MEM-A) Picatinny Arsenal, NJ 07806-5000	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
U.S. Army ARDEC, ESIC Knowledge & Process Management (RDAR-EIK) Picatinny Arsenal, NJ 07806-5000	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A three state Kalman filter was developed to estimate the airspeed of a projectile given only measurements of its axial acceleration, along with a priori knowledge of the axial force coefficient of the projectile. The nonlinear hybrid filter was developed, tested, and tuned using simulated data and its performance on flight data was evaluated.

15. SUBJECT TERMS

Extended, Hybrid, Kalman, Filter, EKF, Velocity, Estimation, Projectile, Axial force coefficient, Accelerometer

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Thomas Recchia
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)
U	U	U	SAR	23	(973) 724-8853

CONTENTS

	Page
Summary	1
Introduction	1
Methods, Assumptions, and Procedures	2
Observability	2
System Dynamics and Measurement Equation	3
Linearization of System	3
Continuous Filter Propagation	6
Discrete Filter Update	7
Testing and Tuning of Filter	8
Results and Discussion	9
Simulated Trajectory	9
Performance of the Filter on Actual Data	13
Conclusions	14
References	15
Distribution List	17

FIGURES

1	Diagram of state variables	3
2	Projectile axial acceleration	4
3	Velocity estimate from perfect initialization	9
4	Altitude estimate from perfect initialization	10
5	Flight path angle from perfect initialization	10
6	Velocity estimate with low muzzle velocity initialization	11
7	Altitude estimate with low muzzle velocity initialization	12
8	Flight path angle with low muzzle velocity initialization	12
9	Filter predicted acceleration versus measured acceleration	13
10	Velocity estimated from flight data compared with radar velocity measurement	14

PREFACE

Guided projectiles generally have a requirement for estimating their current states in order to accurately guide to their targets. While recently it has become more common to accomplish this with GPS, there are many projectiles that continue to employ inertial measurement units (IMU) to measure their linear accelerations and rotational rates. One common way to use this data is to integrate it to get the projectile velocity and position while in flight. The purpose of this report is to propose an alternate approach that allows the projectile to estimate its current airspeed using its measured axial acceleration and a **priori** knowledge of its drag characteristics using a three-state extended hybrid Kalman filter. This airspeed estimate can be used in the guidance system to look up aerodynamic coefficients or to estimate an absolute value of velocity to be used in integrating the accelerations measured by the IMU.

For the purposes of this work, it is assumed that high quality axial accelerometer data is available to the guidance processor, as is often the case on guided munitions. This data is assumed to have negligible bias induced by gun-launch and negligible drift. While bias and drift errors could be addressed in the future, the current work serves as a proof of concept for using the axial acceleration data to estimate airspeed.

The author would like to thank Professor Richard Haddad, without whose insight, encouragement, and time this work would not have been completed.

SUMMARY

In order for guided projectiles to successfully guide themselves to a given target, it is often necessary for them to estimate their current air-relative velocity. In the case of a guided projectile that has an inertial measurement unit (IMU), the axial acceleration measurement may be used to determine the absolute air-relative velocity. This is possible if the axial force coefficient and mass properties of the projectile are known **a priori**.

One way to implement this velocity estimation is with an extended hybrid Kalman filter that takes into account the non-linearities in the system dynamics, while allowing for discrete measurements to be incorporated into the estimates. The filter designed here estimates three states of the projectile, including the air-relative velocity, the altitude, and the flight path angle. The filter was tuned and tested on simulated data first, to evaluate its performance against known values. It was then tested on actual flight data. It was shown to perform well in both of these scenarios

INTRODUCTION

Guided projectiles often take many measurements during flight in order to estimate their current position, velocity, angular position, and angular velocity. These estimates are crucial for the projectile to successfully guide itself to its intended target. These measurements can include global positioning system (GPS) position and velocity data, body-fixed accelerometer data, body-fixed rate sensor data, seeker measurements, etc. The purpose of this report is to provide a way to use axial accelerometer data, along with **a priori** knowledge of the axial force coefficient of the projectile, to estimate the absolute airspeed of a projectile. The solution method proposed uses an extended hybrid Kalman filter to provide this estimate.

The process is assumed to begin with high quality accelerometer data that has a very small bias induced by gun launch and a very low drift rate. Both of these sources of error are assumed to be negligible for the filter that is developed. In addition, it is assumed that the accelerometer data has white, Gaussian noise on top of the signal of interest. At the time of writing, data of this quality were available for investigation of this filter.

This report will first present the dynamic model of the projectile, along with the measurement equation used in the filter. It will then explain the processing required to convert the dynamic model into the required matrices for the Kalman filter, detailing how the non-linearities were handled in determining the Jacobian of the system. Simulation data that was used to tune the filter will then be presented along with a discussion of the filter's performance. Finally, a sample data set from a real test flight will be presented to show the filter's performance on real-world data. The primary reference used for the Kalman filter equations is Chapter 13 of Optimal State Estimation, by Dan Simon (ref. 1).

METHODS, ASSUMPTIONS, AND PROCEDURES

Observability

The most important premise of the development of the Kalman filter is that the state of interest is observable from the measurements available. Since the equations involved are nonlinear and involve table lookups, the standard observability matrix cannot be formed. However, it is possible to make a more intuitive argument that the absolute velocity, relative to the air mass, is observable from the acceleration measurement.

If it is assumed that the projectile flies with a very small angle of attack, it can be assumed that the aerodynamics are only a function of Mach number. In addition, if the projectile is flying a ballistic trajectory, then the only aerodynamic coefficient affecting the axial acceleration is the axial force coefficient, CX. Mach number is defined as the air-relative velocity divided by the speed of sound, which varies in a known way with altitude. Therefore, the measured axial acceleration of the projectile can be written as follows:

$$a_x = \frac{1}{2m} V^2 S_{ref} \rho(z) CX(V,z) \tag{1}$$

where

a_x = measured axial acceleration

m = projectile mass

V = total projectile velocity relative to the air

S_{ref} = reference area of the projectile

ρ = air density

z = altitude

CX = axial force coefficient

Here the reference area of the projectile is determined by how the axial force coefficient was calculated a priori, and it is usually equal to the projected frontal area of the body. In addition, the air density is assumed to vary with altitude. An important note is that there is no gravity term in this equation because although the projectile experiences accelerations due to gravity, the accelerometer proof mass experiences exactly the same accelerations, so gravitational accelerations are not measurable with accelerometers in free flight. From this equation, it is apparent that given knowledge of the projectile geometry, mass, and axial force coefficient along with an estimate of the altitude, an iterative solution method would be able to calculate the air-relative velocity from a measurement of the axial acceleration. Although there are many options, particularly if it is only desired to solve this equation at a single point, the iterative solution method chosen for this work was an extended hybrid Kalman filter. This choice was made because the Kalman filter will also allow the altitude to be estimated throughout the flight.

System Dynamics and Measurement Equation

The first step in developing the filter is to define the system dynamics in a way that captures the important relationships between states, but also is described in terms of the states of interest so they will be estimated directly by the filter. The first state chosen was the air-relative velocity, V. This is the state of interest, so including it directly in the system dynamic equations was natural. From the observability analysis, the altitude, z, also needs to be estimated, so this was included as the second state. In order to propagate these two states, a relationship between them is needed. For this, a third state was added. This state is the flight path angle, theta, which is the angle between the air-relative velocity and the horizontal. The state variables chosen are illustrated in figure 1. With these three states, the system dynamics can be propagated accurately according to the following system of differential equations.

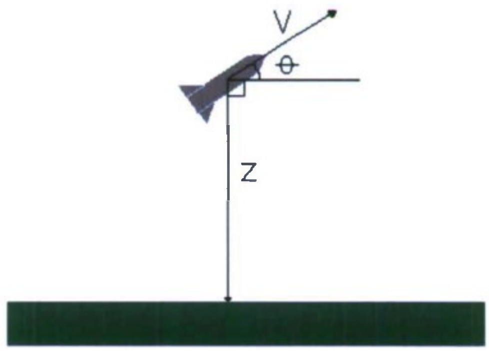

Figure 1
Diagram of state variables

$$\begin{matrix}\dot{V}\\ \dot{z}\\ \dot{\theta}\end{matrix} = f\left(\begin{bmatrix}V\\ z\\ \theta\end{bmatrix}\right) = \begin{cases}\dot{V} = \frac{1}{2m}\rho V^2 S_{ref} CX - g\ sin(\theta)\\ \dot{z} = -V\ sin(\theta)\\ \dot{\theta} = -\frac{g}{V}\cos(\theta)\end{cases} \tag{2}$$

Note that since these equations need to describe the actual dynamics of the system, the acceleration due to gravity is now included. Also, there is an implicit assumption that there is no vertical velocity component of the wind. This means that the derivative of the altitude, z, is directly related to the air-relative velocity V, without knowledge of the wind. Here the explicit dependence of ρ and CX on velocity and altitude are omitted for clarity, but it should be understood that they are implicit functions of these state variables. In addition, the CX function is implemented in the filter as a table lookup with Mach number as the independent variable. This allows the filter to use nonlinear wind tunnel or flight test data for CX.

The next equation needed to develop the Kalman filter is the measurement equation. This is exactly the same as equation 1, with the understanding that this is the theoretical measurement. The Kalman filter development assumes that there is additive white Gaussian noise contained in the actual measurements taken. The measurement equation is repeated here using the Kalman filter notation, h, to represent the measurements taken.

$$h(V,z) = a_x = \frac{1}{2m}\rho V^2 S_{ref} CX \tag{3}$$

Here, the measurement is a function of two of the state variables, V and z. Again, the functional dependence of ρ and CX on V and z are implied. Note that the axial acceleration is assumed to be in the same direction as the velocity. This is illustrated in figure 2, and is a consequence of the assumption that the projectile is flying with a negligible angle of attack and is following a ballistic flight path.

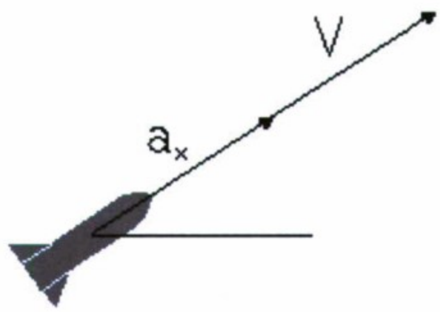

Figure 2
Projectile axial acceleration

Linearization of System

The next step to developing the filter is to compute the Jacobian of the system dynamics and Jacobian of the measurement equations. These matrices will be evaluated at each time step and used by the filter to update the estimates of the state variables. The Jacobian of the system is calculated by taking partial derivatives of each of the state equations with respect to each of the state variables, resulting in a 3 by 3 matrix as follows

$$A = \begin{bmatrix} \frac{\partial f}{\partial V} & \frac{\partial f}{\partial z} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{4}$$

Taking the partial derivatives yields the following values for the elements of this matrix

$$\begin{aligned}
A_{11} &= \frac{1}{m}\rho V S_{ref}\left\{CX + \frac{1}{2}V\left(\frac{\partial CX}{\partial V}\right)\right\} \\
A_{12} &= \frac{1}{2m}V^2 S_{ref}\left\{\rho\left(\frac{\partial CX}{\partial z}\right) + CX\left(\frac{\partial \rho}{\partial z}\right)\right\} \\
A_{13} &= -g\ \cos(\theta) \\
A_{21} &= -\sin(\theta) \\
A_{22} &= 0 \\
A_{23} &= -V\cos(\theta) \\
A_{31} &= \frac{1}{V^2}g\ \cos(\theta)
\end{aligned} \tag{5}$$

$$A_{32} = 0$$

$$A_{33} = \frac{1}{V} g \sin(\theta)$$

Notice that there are still some partial derivatives left to evaluate. This will be addressed shortly, but first, the Jacobian of the measurement equation will be evaluated. In this case, since there is only one measurement equation, the resulting Jacobian will be a 1 by 3 matrix.

$$H = \begin{bmatrix} \frac{\partial h}{\partial V} & \frac{\partial h}{\partial z} & \frac{\partial h}{\partial \theta} \end{bmatrix} = [H_{11} \quad H_{12} \quad H_{13}] \tag{6}$$

where

$$H_{11} = \frac{1}{m} \rho V S_{ref} \left\{ CX + \frac{1}{2} V \left(\frac{\partial CX}{\partial V} \right) \right\}$$

$$H_{12} = \frac{1}{2m} V^2 S_{ref} \left\{ \rho \left(\frac{\partial CX}{\partial z} \right) + CX \left(\frac{\partial \rho}{\partial z} \right) \right\} \tag{7}$$

$$H_{13} = 0$$

Here again, note that there are partial derivatives still present. In fact, these are the same three partial derivatives needed for the evaluation of the system Jacobian. Each of these partial derivatives is evaluated in a different way.

The partial derivative of CX with respect to z was evaluated in a brute force manner. This was necessitated by the fact that CX was input to the filter as a table of CX versus Mach number, which meant that an analytical partial derivative for CX with respect to z was not calculable. In this case, a second lookup of CX' based on a slightly perturbed altitude, z', was performed. A perturbation of 0.01 m worked well for the filter. Then, the partial derivative was evaluated numerically using the following equation

$$\frac{\partial CX}{\partial z} = \frac{CX' - CX}{z' - z} \tag{8}$$

The partial derivative of CX with respect to V was evaluated in a slightly less brute force manner. Although this partial derivative was still evaluated numerically, the bulk of the calculations were done off-line, ahead of time. This was enabled by the fact that the partial derivative of CX with respect to V is simply related to the partial derivative of CX with respect to Mach number, M, as shown in equation 9.

$$\frac{\partial CX}{\partial V} = \frac{1}{a} \left(\frac{\partial CX}{\partial M} \right) \tag{9}$$

Here, a is the speed of sound and can be calculated (ref. 2) from the estimate of z at each time step where this partial derivative needs to be evaluated. The partial derivative of CX with respect to M was calculated numerically, ahead of time, from the table of CX versus Mach number using a simple finite difference equation.

The last expression to evaluate is the partial derivative of ρ with respect to z. This partial derivative was evaluated more analytically. A standard atmosphere model given by McCormick (ref. 2) was differentiated analytically to yield an expression for this partial derivative as given in equation 10 in terms of the density and temperature at ground level, ρ_0 and T_0, respectively

$$\frac{\partial\rho}{\partial z} = 0.0277\left(\frac{\rho_0}{T_0}\right)^{4.256}(T_0 + 0.00651z)^{3.256} \tag{10}$$

The units of temperature are Kelvin, the units of density are kilograms per meter cubed, and the altitude, z, is in meters. This equation is based on a very good fit to tabular data, which is applicable up to approximately 11 km above sea level.

Continuous Filter Propagation

For each time step of the filter, the states and error covariance matrix are first propagated according to the system dynamic model and the system dynamic noise. Since this filter is nonlinear, the easiest way to do this was to use a numerical integration scheme to advance the filter in time. The time step for the integration was set to match the data rate of the IMU data. A Runge-Kutta 4th order integration (ref. 3) was carried out to advance the system states as follows

$$\begin{aligned}
&x = [V \quad z \quad \theta]^T \\
&dt = t_k - t_{k-1} \\
&f_1 = f(x_{k-1}^+, t_{k-1}) \\
&f_2 = f\left(x_{k-1}^+ + f_1\frac{dt}{2}, t_{k-1} + \frac{dt}{2}\right) \\
&f_3 = f\left(x_{k-1}^+ + f_2\frac{dt}{2}, t_{k-1} + \frac{dt}{2}\right) \\
&f_4 = f(x_{k-1}^+ + f_3, t_{k-1} + dt) \\
&x_k^- = x_{k-1}^+ + \frac{1}{6}(f_1 + 2f_2 + 2f_3 + f_4)dt
\end{aligned} \tag{11}$$

Note here the switch to typical Kalman filter type notation. The subscript refers to the time index. The superscript plus means that the state was updated by the available measurement. The superscript minus means that the state was advanced in time by the system dynamics, but it has not been updated yet. So the purpose of this step is to advance from the last updated state to the next predicted state.

The other matrix that needs to be advanced in time is the error covariance matrix, P. This matrix contains the filter's estimated error covariances for the states being estimated. During the propagation step this matrix grows depending on the system dynamics Jacobian as well as the system dynamic noise, Q. The system dynamic noise is a way to tell the filter how much the system dynamic model should be trusted. The relationship used allows the time derivative of P to be calculated based on A and Q. The error covariance matrix, P, is then propagated using a 1st order Eulerian integration scheme as follows

$$\dot{P}_k = AP_{k-1}^{+} + P_{k-1}^{+}A^T + Q$$
$$P_k^{-} = P_{k-1}^{+} + \dot{P}_k dt \tag{12}$$

Note that again, the filter is propagating the previous step's updated estimate to get the current step's predicted value of the error covariance matrix. This matrix will be updated in the next step, along with the system dynamics, to account for the measurements being taken.

Discrete Filter Update

Since the measurements are being taken discretely, it is natural to use the discrete Kalman filter equations (ref. 1) to update the state estimates and error covariance matrix. The fact that the filter includes continuous propagation and discrete update makes it a hybrid filter. The first update step is to calculate the Kalman gain, K, from the error covariance matrix, the measurement Jacobian, and the measurement covariance matrix, R. Because this system has only one measurement, the measurement covariance matrix is a scalar that is the variance of the measured acceleration. The Kalman gain matrix is calculated as follows

$$K = P_k^{-}H^T(HP_k^{-}H^T + R)^{-1} \tag{13}$$

The next step is to update the filter estimates using the measured acceleration, a_{meas}, the measurement equation, and the Kalman gains. This step produces the best estimate at the current time of the states of the system.

$$x_k^{+} = x_k^{-} + K\left(a_{meas} - h(x_k^{-})\right) \tag{14}$$

The final step is to update the error covariance matrix, P, using the Kalman gains, the measurement Jacobian, and the measurement covariance matrix, R. This step estimates the performance of the filter at the current time step.

$$P_k^{+} = (I - KH)P_k^{-}(I - KH)^{-1} + KRK^T \tag{15}$$

At this point the filter equations are complete. Using these equations, the filter can be propagated and updated to provide the best estimates of the states and the estimate of its own performance in terms of the error covariance matrix, throughout the trajectory.

Testing and Tuning the Filter

The filter was tested and tuned using a three degree of freedom (3DOF) point-mass simulated trajectory that was representative of the intended application. The simulated total velocity and the simulated axial acceleration due to aerodynamic forces were input to the filter. White Gaussian noise was added to the axial acceleration to generate a simulated measurement. It is important to note that to isolate the performance of the filter, the trajectory simulation and the Kalman filter had exactly the same piecewise function for CX as a function of Mach number. This is important because the filter is sensitive to errors in the CX function, and the filter was tuned with an essentially perfect model for the simulated acceleration measurements. The filter was subsequently tested on flight test data that included errors in the CX function.

In order to run the filter on this data, several inputs are needed that adjust how the filter performs. The first input to the filter was the initial error covariance matrix. This matrix tells the filter how confident it should be in the start values of the state estimates. Since the firing platform for the projectile knows its initial altitude and quadrant elevation relatively accurately, the starting error covariances for these two states were small. Since the muzzle velocity was known with relatively little confidence, the error covariance for it was set much higher than the other two. The actual error covariance matrix input to the filter was

$$P_0^+ = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.001 \end{pmatrix} \tag{16}$$

The next input to the filter was the dynamic model noise covariance matrix, Q. This matrix defines for the filter the expected error in the model from the true system dynamics. In practice, it is used to adjust the performance of the filter by telling it to trust the dynamics more or less as it acquires data. For this case, the filter was set to trust the filter altitude and flight path angle models because they did not affect the estimate of velocity very strongly. The filter predictions for these two states are good enough that even if they vary from the truth, the estimate for the velocity is still excellent. The dynamic model noise covariance matrix was set as follows

$$Q = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix} \tag{17}$$

The last input to the filter was the measurement covariance matrix, R. As mentioned previously, this is only a scalar variance for the filter in question, because there is only one measurement being taken. For this filter, R was set to 0.25. This value corresponds to the expected variance of the measurement device, in this case the axial accelerometer. This value was estimated from previous data available from the device on the projectile, and it was used to simulate the noise in the simulated acceleration data.

RESULTS AND DISCUSSION

Simulated Trajectory

The first step was to run the filter under perfect conditions. In this case, the states were initialized perfectly to match the simulation. The initial velocity was 490 m/s, the initial altitude was 0 m, and the initial flight path angle was 30 deg. The filter initialization parameters were tuned to the values reported previously, until the filter gave a good estimate of the velocity. Figures 3 to 5 show the results of running the filter with perfect initialization. The red curves represent the true values from the simulation. The blue curves represent the filter estimates of the velocity, altitude, and glide angle, respectively.

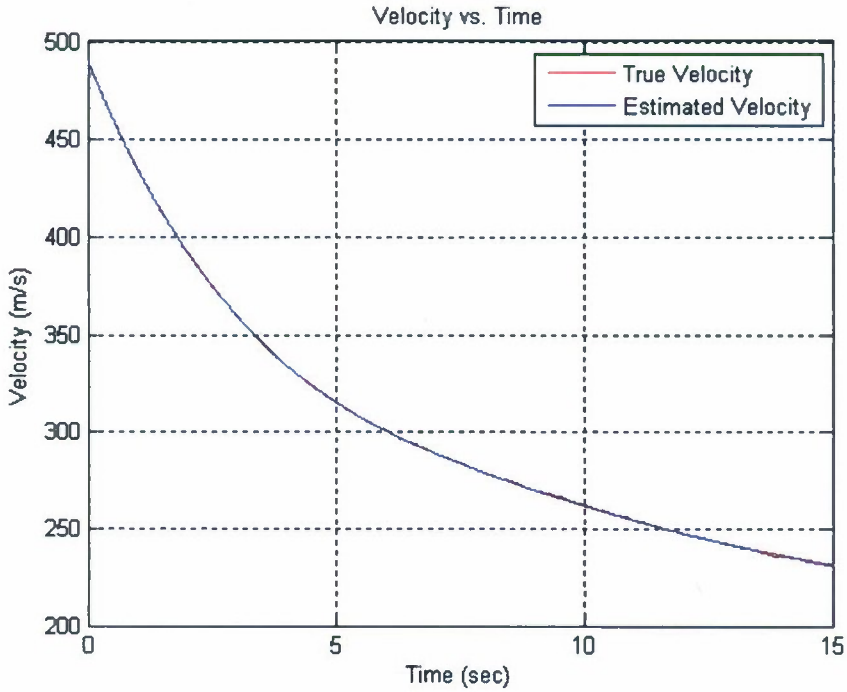

Figure 3
Velocity estimate from perfect initialization

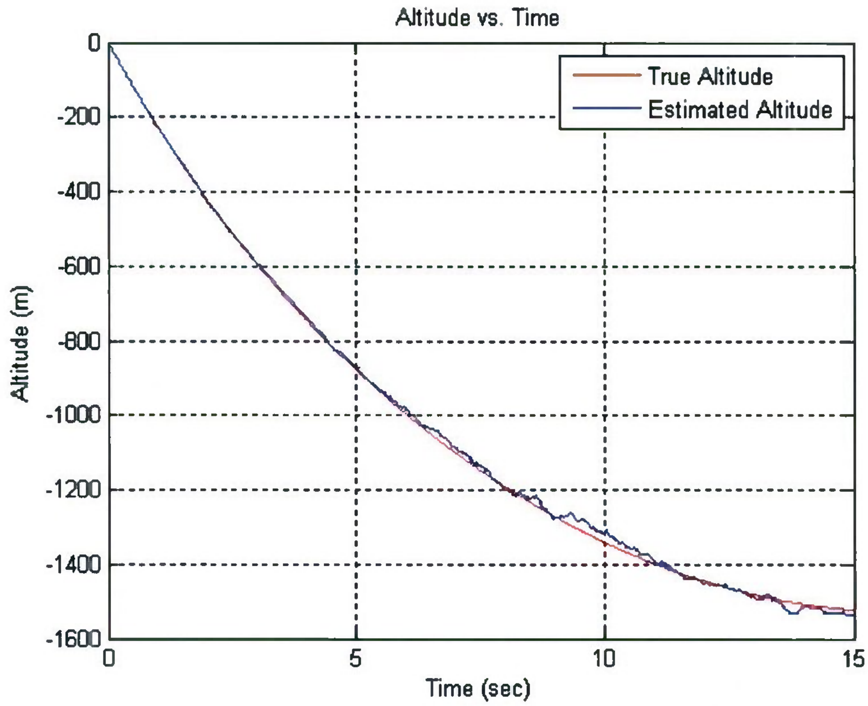

Figure 4
Altitude estimate from perfect initialization

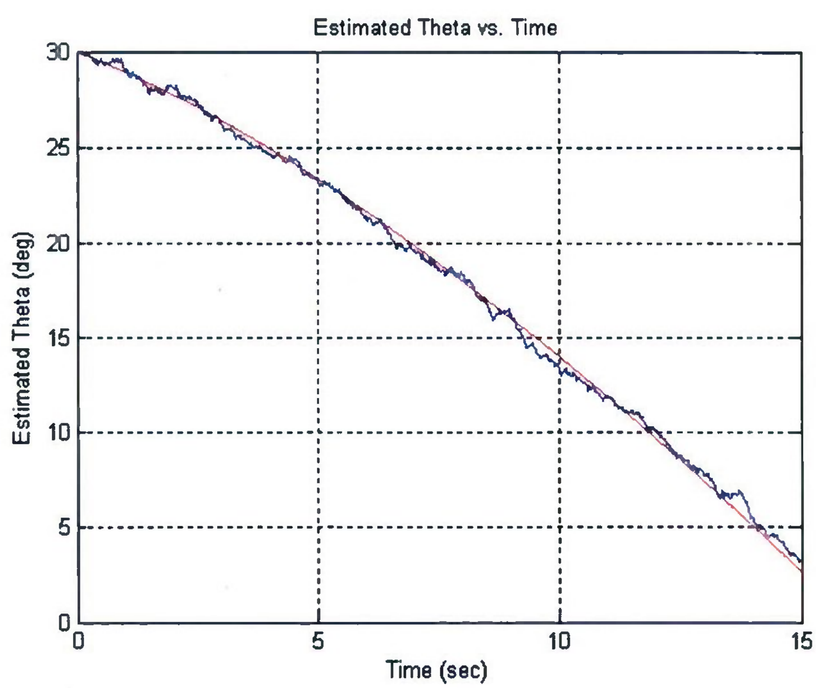

Figure 5
Flight path angle from perfect initialization

It is apparent from these plots that the filter is doing an excellent job of estimating all three states of the system. In fact, the velocity estimate is so good that it is difficult to see the truth curve since it is covered by the estimate. This is expected, since the dynamic model in the filter is equivalent to the dynamic model used by the simulation to produce the trajectory in the first place.

The next step was to evaluate the performance of the filter under less than ideal conditions. The state that is least well known at gun launch is the muzzle velocity. This is due to the natural variation in propellant burn rate at different bed temperatures and differing propellant performance as it ages. The initial flight path angle is well known as the quadrant elevation to which the gun was set. In addition, the initial altitude of the firing platform is well known from instruments and maps available to the vehicle or firing crew. So the most likely stress case for this filter was a large error in initial muzzle velocity. It was chosen to initialize the filter with a muzzle velocity of 200 m/s. This is 290 m/s slower than the actual simulated muzzle velocity of 490 m/s. In addition, it stresses the filter further because the CX function is not well behaved at Mach equal to 1 and the filter was initialized in the subsonic regime, while the actual velocity was supersonic. The results of this run are shown below in figures 6 to 8.

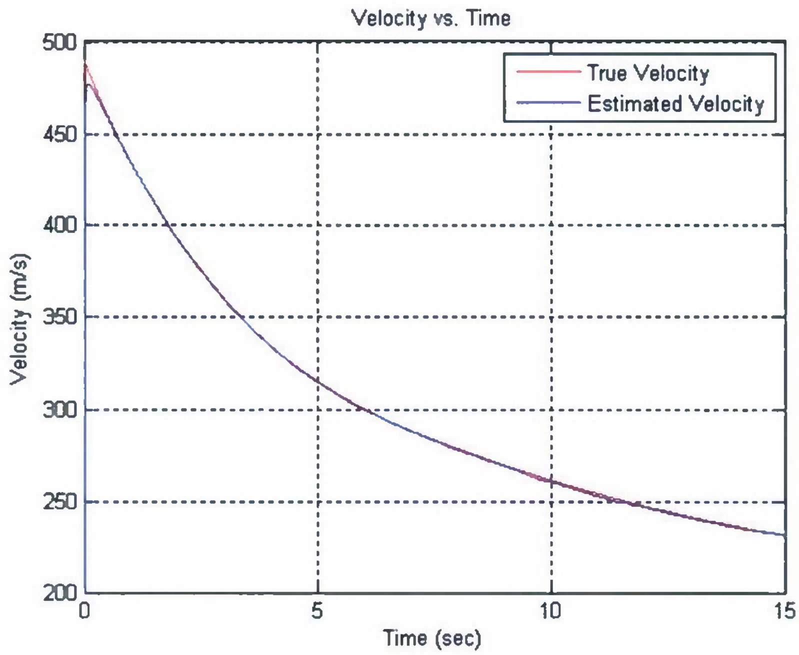

Figure 6
Velocity estimate with low muzzle velocity initialization

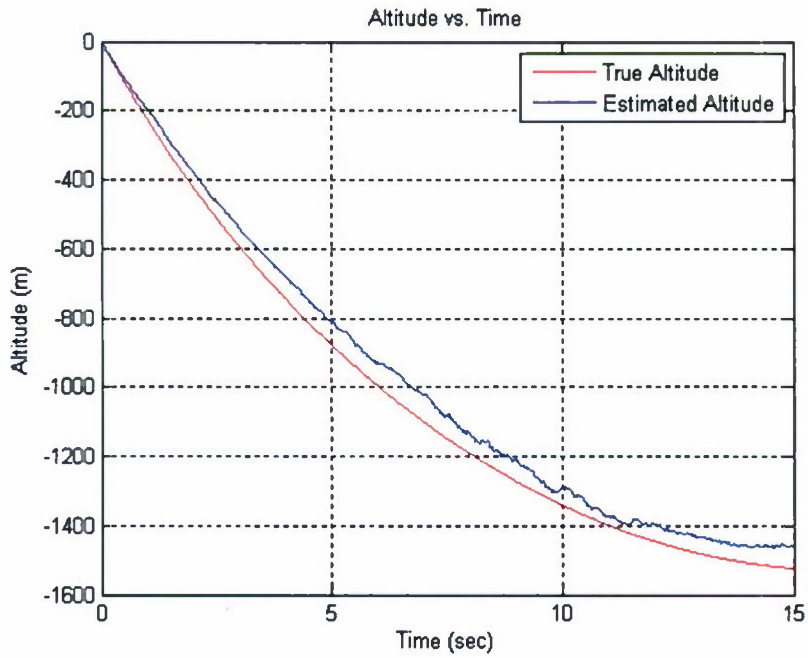

Figure 7
Altitude estimate with low muzzle velocity initialization

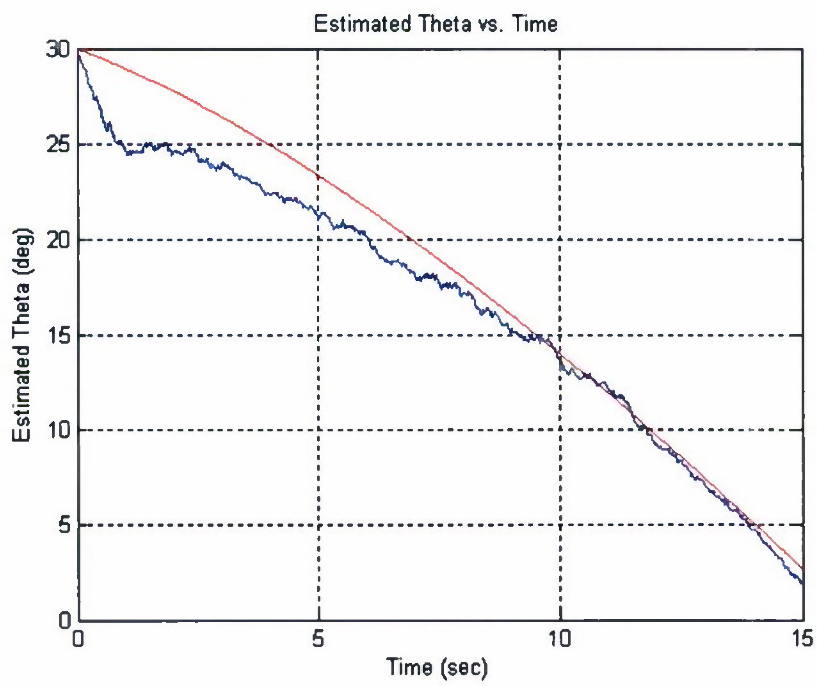

Figure 8
Flight path angle with low muzzle velocity initialization

In this case, the filter converges very quickly to an accurate estimate of the velocity of the projectile, even though the initial estimate was far from the actual value. After it converges, the error is less than 1 m/s between the estimate and the truth value for the rest of the flight. However, the estimates for the altitude and flight path angle have more significant errors in them. This is due to how the dynamic model noise covariance matrix was set up. It is acceptable performance for this case because the projectile does not fly very high, and the overall variations in density and CX due to altitude are small to begin with. The error in the estimate of theta is accounted for already in the altitude estimate, so as long as the velocity is estimated with satisfactory accuracy, the filter has done its job.

Performance of the Filter on Actual Data

The next step of evaluating the filter performance was to run it on actual flight data from a projectile flying with an axial accelerometer. For this case, the CX model used was calculated from the accelerometer data itself based on measured projectile mass properties and known meteorological data taken during the test. This is not how the filter would be run in practice, because in that case CX would need to be known a **priori**. However, it does provide a test case to show how the filter operates on real data with real noise in it. This is an important test because the noise was assumed to be white Gaussian noise and the bias and drift rates were assumed to be negligible, but that was never verified. In addition, the test data included the effects of a real atmosphere–wind, non-ideal variations with altitude, etc. This test case provides information on how the filter might perform in a real tactical scenario.

Figure 9 shows the projectile acceleration predicted in the filter displayed in blue over the red raw accelerometer flight data. The filter tracks the measurement quite accurately, smoothing out the noise. This plot is a good check to make sure that the filter predicted measurement is tracking the actual measurement. This means the filter dynamics and theoretical measurement equations are accurate enough to model the real scenario.

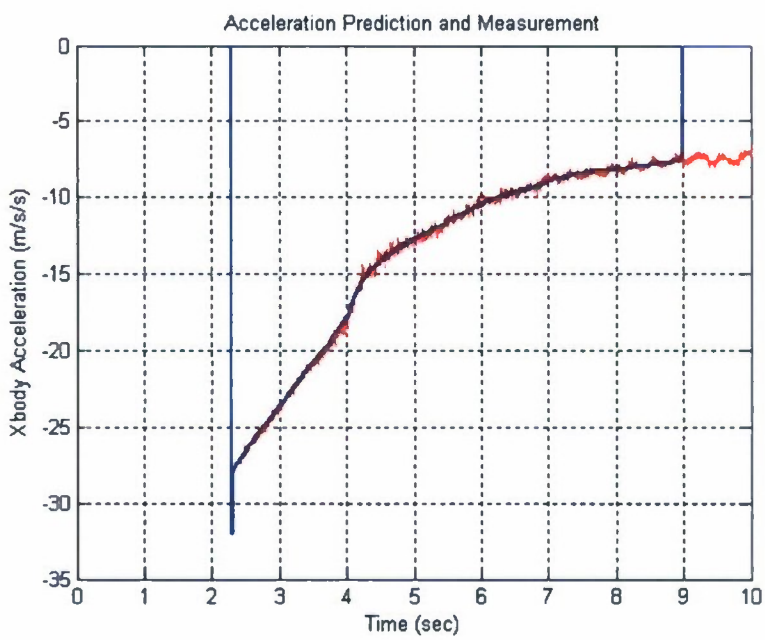

Figure 9
Filter predicted acceleration versus measured acceleration

Figure 10 shows the filtered estimated velocity in blue over the measured radar total velocity in red. The filter tracks the actual velocity quite well. The differences between these two curves may come from the fact that the filter estimates air-relative velocity and radar measures ground-relative velocity. In addition, the accelerometer data starts a little after 2 sec into the flight. Since this was a flight test, the true states are not available, so the filter was initialized with an approximate guess–the initial expected velocity, gun altitude, and gun quadrant elevation. Errors in these initializations may have contributed somewhat to the errors between the estimated velocity and the radar velocity measurement. It is worth noting that the filter executed faster than real-time on a Dell Precision 690 3.00 GHz processor, in a single thread Matlab instance. Although not guaranteed, this means the filter is probably small enough to run on flight hardware in real-time.

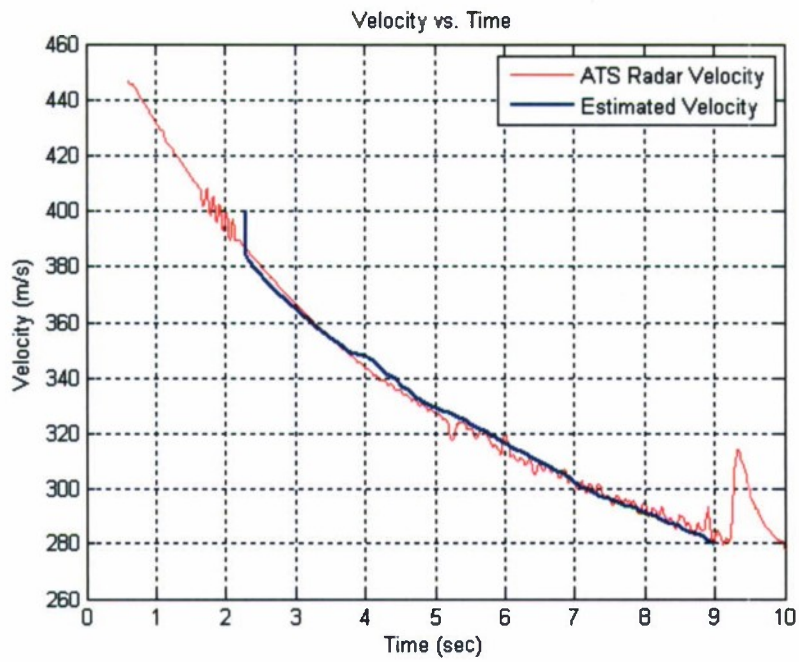

Figure 10
Velocity estimated from flight data compared with radar velocity measurement

CONCLUSIONS

In order to estimate air-relative velocity purely from accelerometer measurements and a priori knowledge of projectile characteristics, an extended hybrid Kalman filter can be used. This simple filter requires only three states to provide satisfactory performance in estimating projectile velocity. However, the filter is sensitive to errors in the knowledge of the axial force coefficient, and the air-relative velocity must be corrected with the wind velocity in order to measure ground-relative velocity. The air-relative velocity estimate may be used by other guidance algorithms on the projectile for aerodynamic lookup, while the ground-relative velocity might be used for navigation purposes.

REFERENCES

1. Simon, Dan, Optimal State Estimation, John Wiley & Sons, Inc., Hoboken, NJ, 2006.

2. McCormick, Barnes W., Aerodynamics Aeronautics and Flight Mechanics, 2nd ed., John Wiley & Sons, Inc., New York, NY, 1995.

3. Mathews, John H. and Fink, Kurtis D., Numerical Methods using Matlab, 3rd ed., Prentice-Hall, Inc., 1999.

DISTRIBUTION LIST

U.S. Army ARDEC
ATTN: AMSRD-AAR-EIK
AMSRD-AAR-GC
RDAR-MEM, E. Logsdon
RDAR-MEM-A, T. Recchia (5)
J. Grau
S. Chung
W. Koenig
A. Farina
E. Vazquez
W. Toledo
G. Malejko
D. Pedersen
Y. Chen
K. Kwok
RDAR-MEF-I, S. Longo
C. Sandberg
M. Heiskanen
J. Choi
L. Vo
RDAR-MEF-S, D. Panhorst
N. Gray
R. Fullerton
B. DeFranco
J. Sarruda
M. Marsh
J. Carson
G. Miner
RDAR-WSH-N, J. Chung
RDAR-MEM-L, A. Vella
RDAR-EIS-AM, J. Walsh
RDAR-MEM-M, J. Travaille
RDAR-MEM-C, P. Magnotti
SFAE-AMO-MAS-LC, D. Rigoglioso
J. McNaboe
SFAE-AMO-MAS-SETI, J. Foultz
SFAE-AMO-CAS, M. Moratz
P. Manz
Picatinny Arsenal, NJ 07806-5000

Defense Technical Information Center (DTIC)
ATTN: Accessions Division
8725 John J. Kingman Road, Ste 0944
Fort Belvoir, VA 22060-6218

Commander
Soldier and Biological/Chemical Command
ATTN: AMSSB-CII, Library
Aberdeen Proving Ground, MD 21010-5423

Department of the Army
U.S. Army Research Laboratory
ATTN: AMSRL-CI-LP, Technical Library
RDRL-WML-E, P. Weinacht
G. Cooper
F. Fresconi
RDRL-WML-F, D. Lyon
D. Hepner
T. Harkins
Mark Ilg
Jim Maley
Bldg 4600
Aberdeen Proving Ground, MD 21005-5066

Chief
Benet Weapons Laboratory, CCAC
Armament Research, Development and Engineering Center
U.S. Army Research, Development and Engineering Command
ATTN: RDAR-WSW
Watervliet, NY 12189-5000

Director
U.S. Army TRADOC Analysis Center-WSMR
ATTN: ATRC-WSS-R
White Sands Missile Range, NM 88002

Chemical Propulsion Information Agency
ATTN: Accessions
10630 Little Patuxent Parkway, Suite 202
Columbia, MD 21044-3204

GIDEP Operations Center
P.O. Box 8000
Corona, VA 91718-8000